

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Application Serial No.: 10/849,294  
Applicants: Plante  
Filing Date: May 19, 2004  
Art Unit: 2192  
Examiner: Wei, Zheng  
Confirmation No.: 5608  
Title: Method for Synchronization of Concurrently Modified  
Interdependent Semi-Derived Artifacts

Commissioner for Patents  
Mail Stop Appeal Brief - Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**APPEAL BRIEF**

In response to the final Office Action, dated June 22, 2009, rejecting pending claims 1-4, and in support of the Notice of Appeal received by the U.S. Patent and Trademark Office on August 10, 2009, Appellant hereby submits this Appeal Brief to the Board of Patent Appeals and Interferences. Authorization is herein granted to apply the \$540.00 requisite fee set forth in 37 C.F.R. §41.20(b)(2), and any other fees or credits due in this case to Deposit Account No. 122158. Appellant respectfully requests reconsideration and reversal of the Examiner's rejections of the pending claims.

## TABLE OF CONTENTS

Real Party in Interest .....	3
Related Appeals and Interferences .....	3
Status of Claims .....	3
Status of Amendments .....	4
Summary of Claimed Subject Matter .....	4-5
Grounds of Rejection to be Reviewed on Appeal .....	5
Argument .....	6-11
Claims Appendix .....	12-13
Evidence Appendix .....	14
Related Proceedings Appendix .....	14

REAL PARTY IN INTEREST

The Real Party in Interest is International Business Machines Corporation, the owner of all rights of this patent application by virtue of an assignment recorded at reel and frame number 014994/0182.

RELATED APPEALS AND INTERFERENCES

None.

STATUS OF CLAIMS

The patent application as originally filed included claims 1-18. Claims 1, 7 and 15 were amended and claims 5, 6, 9-14, 17 and 18 were canceled in an Amendment and Response dated June 13, 2007 in response to a non-final Office Action mailed April 17, 2007. Claims 3 and 4 were amended in an Amendment and Response dated November 16, 2007 in response to a non-final Office Action mailed August 22, 2007. Claim 1 was amended and claims 7, 8, 15 and 16 were canceled in an Amendment and Response dated April 14, 2008 in response to a final Office Action mailed February 7, 2008. A Request for Reconsideration in which no claim amendments were made was filed October 6, 2008 in response to a non-final Office Action mailed July 17, 2008. A Request for Reconsideration in which no claim amendments were made was filed March 4, 2009 in response to a non-final Office Action mailed January 12, 2009. The final Office Action mailed June 22, 2009 rejects claims 1-4. Claims 1-4 remain pending in the application and are the subject of this appeal.

### STATUS OF AMENDMENTS

No amendments or other forms of response have been filed subsequent to the mailing of the final Office Action mailed June 22, 2009.

### SUMMARY OF CLAIMED SUBJECT MATTER

#### Independent claim 1

Appellant's invention, as recited in claim 1, features a method for synchronizing a first artifact and a second artifact. (For example, the first and second artifacts can be a software modeling UML file and a third-generation language (3GL) code file. See page 8, line 11 to page 9, line 5.) The first and second artifacts are interdependent. See page 9, lines 2-5. Each artifact 14 and 18 is modified independent of a modification to the other artifact after a last synchronization. See the independently-generated versions of the artifacts in the timeline shown in FIG. 1 subsequent to the single depicted synchronization (SYNC) with descriptive text at page 9, lines 6-16. Also see the independently-generated versions of the artifacts in the timelines shown in FIG. 2 and FIG. 3 subsequent to the first synchronization but before a second synchronization. Each of the first and second artifacts has a plurality of elements and is of a different format. See page 8, lines 11 and 12 (artifact is a file); page 10, lines 1 and 2 (artifacts have elements); and page 8, line 12 to page 13, line 5 (UML and 3GL artifacts or other related artifacts that require application of a forward engineering operation or a reverse engineering operation to transform from one format to the other format).

A reverse engineering operation is performed to generate a temporary artifact having all the elements of a last synchronized version of the first artifact and having all the elements of a

latest version of the second artifact transformed as the first artifact. See step 110 in FIG. 3 and page 10, line 15 to page 11, line 2. Also see FIG. 5 and page 11, lines 3-14 describing how the particular limitations recited for performing the claimed reverse engineering operation are different from a conventional reverse engineering operation. The temporary artifact and a latest version of the first artifact are merged to create a synchronized version of the first artifact. See step 140 in FIG. 3 and page 11, lines 15-19. A forward engineering operation is performed to generate a synchronized version of the second artifact that has all the elements of the latest version of the second artifact and that also has all the elements of the synchronized version of the first artifact transformed as the second artifact. See step 150 in FIG. 3 and page 12, lines 3-6. Also see FIG. 6 and page 12, lines 10-20 describing how the particular limitations recited for performing the claimed forward engineering operation are different from a conventional forward engineering operation.

#### GROUND OF REJECTION TO BE REVIEWED ON APPEAL

The final Office Action issued the following rejections:

- I. Claims 1, 2 and 4 are rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 6,502,239 B2.
- II. Claim 3 is rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,502,239 B2 in view of U.S. Patent No. 6,038,393.

The grounds of rejection to be reviewed on appeal are grounds I and II as applied to claims 1-4.

## ARGUMENT

### Rejection under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 6,502,239 B2

The final Office Action rejects claims 1, 2 and 4 under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 6,502,239 B2 to Zgarba et al. (hereinafter “Zgarba”).

By way of background, Zgarba discloses a method of round-trip engineering source code from a software model. In particular, Zgarba discloses forward engineering applied to previously reverse engineered code in the software model so that the updated source code generated by the forward engineering does not include any changes to the code that were not made in the software model. (See, e.g., Abstract.) “Importantly, existing source code not affected by changes in the software model is left unchanged.” (Col. 7, lines 4-6)

### Claim 1

Zgarba discloses round-trip software engineering in which a software application is reverse engineered into a software model, the software model is changed, and then the code represented by the software model is re-coded. (Col. 1, lines 5-10) Zgarba addresses how to perform the round-trip engineering that allows the software model to be kept synchronized with the source code (or equivalent objects) without the use of code markers which would otherwise clutter the software code. (Col. 2, lines 9-12) Zgarba stresses the value of “proper round trip engineering of a software project:

Importantly, when a software project is reverse engineered from source code, and the software model generated is forward engineered, the resultant code should be essentially the same, regardless of what information in the source code is represented in the software model. This allows for proper “round-trip” engineering of the software project, so that the software model and the source code can be kept synchronized, updates in both the source code and the software model will be maintained when generating the source code from the software model and vice versa, and comments and other features of the source code not represented in the software model do not get moved around or even disappear in the round-trip process. (Col. 4, lines 18-30)

Appellant’s independent claim 1 relates to a method for synchronizing a first artifact and a second artifact. Claim 1 recites, in part, “each artifact being modified independent of a modification to the other artifact after a last synchronization.”

The final Office Action refers to FIG. 1 and related text, and also col. 1, lines 43-46 to show this limitation. Appellant respectfully disagree because the limitation is not disclosed in the cited text and figures, or elsewhere in Zgarba. FIG. 1 and the corresponding text simply relate to a round-trip engineering process that is used for synchronization in a typical sense in which (1) a change is made to the software model and the software code is then synchronized to the changed software model or (2) a change is made to the software code and the software model is then synchronized to the changed software code. However, Zgarba does not address, either in the cited figure and text or elsewhere in the disclosure, a situation in which both artifacts (i.e., the software model and the software code) are changed independently between synchronizations (i.e., “concurrent modified”) and how to subsequently synchronize the changed software model and the changed software code. Thus Zgarba does not teach or suggest “each artifact being modified independent of a modification to the other artifact after a last synchronization.”

Claim 1 further recites “performing a reverse engineering operation to generate a temporary artifact having all the elements of a last synchronized version of the first artifact and having all the elements of a latest version of the second artifact transformed as the first artifact.” The final Office Action on page 4 equates Appellant’s recited temporary artifact with Zgarba’s meta-model. The final Office Action refers to col. 5, lines 3-5 which states “parsing the code and transforming all the elements of the code which can be represented by the software model into a generic meta-model ...” and col. 6, lines 30-33 which states “If a software model 2 is already present when the data from the source file is imported into the software model, and data from the source file is to be merged into the software model rather than replacing it ....”

Appellant points out that the generic meta-model disclosed in Zgarba should not be interpreted to be the temporary artifact recited in claim 1. According to the claim limitation, the temporary artifact has elements from a version of the first artifact and elements from a version of the second artifact transformed as the first artifact.” Consequently, the elements of Appellant’s recited temporary artifact are not the same as elements that are stored in Zgarba’s generic meta-model. Elements in the generic meta-model are in a format that is different from the formats of the software code and the software model. More specifically, Zgarba describes the generic meta-model as an intermediate database that can hold a representation of the code using, for example, the CASE Data Interchange Format (CDIF). (Col. 5, lines 2-18) Thus Appellant respectfully

submits that Zgarba fails to teach or suggest “performing a reverse engineering operation to generate a temporary artifact having all the elements of a last synchronized version of the first artifact and having all the elements of a latest version of the second artifact transformed as the first artifact.”

With respect to “merging the temporary artifact and a latest version of the first artifact to create a synchronized version of the first artifact” as recited in claim 1, the final Office Action refers to col. 7, lines 2-5 and col. 4, lines 35-45. Appellant notes that col. 7, lines 2-5 simply describes events that have an affect on the results of a forward engineering operation. Appellant also notes that col. 4, lines 35-45 describes a standard merging process that does not rely on Appellant’s recited temporary artifact or an equivalent.

With respect to “performing a forward engineering operation to generate a synchronized version of the second artifact having all the elements of the latest version of the second artifact and having all the elements of the synchronized version of the first artifact transformed as the second artifact” as recited in claim 1, the final Office Action refers to step 10 of FIG. 4 and col. 3, line 67 to col. 4, line 3. Appellant respectfully notes that the identified figure and text simply describe a reverse engineering operation in a general sense; however, no teaching or suggestion is presented to show the reverse engineering being applied to artifacts that are independently modified after a last synchronization as described above and recited earlier in the claim.

Notwithstanding the above arguments, Appellant here addresses the Examiner’s arguments set for in the “Response to Arguments” section on pages 2 and 3 of the final Office Action. Specifically, the Examiner states:



As Zgarba disclosed (col.4, lines 22-27), "This allows for proper 'round-trip' engineering of the software project, so that the software model and the source code can be kept synchronized, updates in both the source code and the software model will be maintained..." [emphasis added]. Moreover, Zgarba further discloses "the existing source code 4 is merged with the data from the software model 2 whereby to generated new source code 8 which then replaces the existing source code" [emphasis added] (see for example, col.4, lines 35-38). That is to say "the data from the software model" (model changed) and "the existing source code" can be merged to generate the new source code, wherein "the existing source code" includes "any parts of the source code not represented by the software model" (modified source code)(see for example, col.4, lines 40-45). Therefore, with the "round-trip", any independently changed/modified artifacts can be synchronized by merging in the same format either in source code format or software model format as disclosed by Zgarba.

Appellant addresses each statement herein in order. First, Appellant notes that nowhere in Zgarba is it taught or suggested that the updates in the source code and the software model are made independently between synchronizations. Next, merging the existing source code with the data from the software model to generate new source code to replace the prior source code is a standard forward engineering application. Appellant's invention as recited in claim 1 is not limited to this forward engineering operation. Finally, the existing source code might be expected to include part of the source code that are not represented by the software model as the model and code generally do not have a one-to-one correspondence between their parts. Such differences are not an indication that something has been changed to cause a lack of correspondence or that both artifacts have been changed independent of each other since the last synchronization.

In summary, Appellant again points out that Zgarba addresses a round-trip process to synchronize a software model and code so that changes made to one are reflected in the other. In contrast, Appellant's invention as recited in method claim 1 is based on processing artifacts that are both modified, independent from each other, between synchronizations. For the reasons set forth above, Appellant respectfully submits that Zgarba does not teach or suggest all the limitations recited in claim 1 and therefore that claim 1 is allowable over Zgarba.

#### Claims 2 and 4

Claims 2 and 4 depend from allowable claim 1 and incorporate all of the limitations of claim 1. Thus claims 2 and 4 are allowable over Zgarba for at least the reasons provided above with respect to claim 1.

#### Rejection under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,502,239 B2 in view of U.S. Patent No. 6,038,393

The final Office Action rejects claim 3 under 35 U.S.C. § 103(a) as being unpatentable over Zgarba in view of U.S. Patent No. 6,038,393 to Iyengar et al. (hereinafter "Iyengar").

#### Claim 3

The final Office Action relies on Iyengar to show a limitation introduced in dependent claim 3 that is not taught or suggested by Zgarba. Claim 3 depends from allowable claim 1 and incorporates all of the limitations of claim 1. The disclosure of Iyengar does not cure the deficiencies noted above with respect to the limitations in claim 1 that are not taught or suggested by Zgarba. Accordingly, claim 3 recites limitations that are not taught or suggested by the cited references, whether taken alone or in proper combination. Thus Appellant respectfully submits that claim 3 is patentable over Zgarba and Iyengar.

In view of the arguments made herein, Appellant submits that the application is in condition for allowance.

Respectfully submitted,

Date: October 1, 2009  
Reg. No. 41,047

Tel. No.: (508) 303-2003  
Fax No.: (508) 303-0005

/William G. Guerin/  
William G. Guerin  
Attorney for Appellant  
Guerin & Rodriguez, LLP  
5 Mount Royal Avenue  
Marlborough, MA 01752

CLAIMS APPENDIX

1. A method for synchronizing a first artifact and a second artifact, the first and second artifacts being interdependent and each artifact being modified independent of a modification to the other artifact after a last synchronization, the first and second artifacts each having a plurality of elements and being of different formats, the method comprising:

performing a reverse engineering operation to generate a temporary artifact having all the elements of a last synchronized version of the first artifact and having all the elements of a latest version of the second artifact transformed as the first artifact;

merging the temporary artifact and a latest version of the first artifact to create a synchronized version of the first artifact; and

performing a forward engineering operation to generate a synchronized version of the second artifact having all the elements of the latest version of the second artifact and having all the elements of the synchronized version of the first artifact transformed as the second artifact.

2. The method of claim 1 wherein one of the first and second artifacts is a software model artifact and the other of the first and second artifacts is a code artifact.

3. The method of claim 2 wherein the software model artifact is a Unified Modeling Language file.

4. The method of claim 2 wherein the code artifact is a third-generation language source file.

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.